

Amendments to the Claims

1-18. (Canceled)

19. (Currently Amended) A method for processing bundled instructions through execution units of a processor, comprising the steps of:

determining a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode;

in the throughput mode:

fetching a first bundle of instructions from a first thread of a multiply-threaded program;

distributing the first bundle to a first cluster of the execution units for execution therethrough;

fetching a second bundle of instructions from a second thread of the program;

distributing the second bundle to the first cluster of the execution units for execution therethrough;

fetching a third bundle of instructions from a third thread of the program;

distributing the third bundle to a second cluster of the execution units for execution therethrough;

fetching a fourth bundle of instructions from a fourth thread of the program; and

distributing the fourth bundle to the second cluster of the execution units for execution therethrough;

wherein the first thread, the second thread, the third thread, and the fourth thread are executing simultaneously; and

in the wide mode:

fetching a fifth bundle of instructions from a fifth thread of the program;

distributing the fifth bundle to only the first cluster for execution therethrough;

fetching a sixth bundle of instructions from the fifth thread of the program; and

distributing the sixth bundle to only the second cluster for execution therethrough.

20. (Previously Presented) The method of claim 19, further comprising:
processing the first, second, and fifth bundles within the first cluster;
processing the third, fourth, and sixth bundles within the second cluster.

21. (Previously Presented) The method of claim 19, further comprising the step of
architecting data from the first cluster to a first register file.

22. (Previously Presented) The method of claim 19, further comprising the step of
architecting data from the second cluster to a second register file.

23. (Previously Presented) The method of claim 19, the steps of fetching the first
through sixth bundles each comprising decoding instructions into the first through sixth bundles,
respectively.

24-26. (Canceled)

27. (Previously Presented) The method of claim 19, further comprising:
bypassing data between the first cluster and the second cluster, as needed, to facilitate the
processing of the fifth bundle through the first cluster and the sixth bundle through the second
cluster.

28. (Previously Presented) The method of claim 27, wherein the step of bypassing the
data utilizes a latch to couple the data between a register file of the first cluster and a register file
of the second cluster.

29. (Previously Presented) The method of claim 19, wherein the step of determining the
mode of operation comprises determining a state of a configuration bit.

30. (Canceled)

31. (Currently Amended) A processor, comprising:

a first cluster and a second cluster, wherein each of the first cluster and the second cluster comprises a plurality of execution units, wherein each of the execution units is configured to process instructions;

a configuration bit configured to specify a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode; and

a thread decoder configured to group instructions of a multiply-threaded program into singly-threaded bundles and to distribute the bundles to the first cluster and the second cluster according to a state of the configuration bit;

wherein during the throughput mode, the thread decoder is configured to distribute the bundles of a first thread and a second thread to the first cluster for processing, and to distribute the bundles of a third thread and a fourth thread to the second cluster for processing, wherein the first thread, the second thread, the third thread, and the fourth thread are executing simultaneously; and

wherein during the wide mode, the thread decoder is configured to distribute each of the bundles of a fifth thread to exactly one of the first cluster and the second cluster for processing.

32. (Previously Presented) The processor of claim 31, wherein each of the first cluster and the second cluster comprises a register file.

33. (Previously Presented) The processor of claim 32, further comprising a latch configured to bypass data between the register file of the first cluster and the register file of the second cluster, as needed, to facilitate the processing of the bundles of the fifth thread.

34. (Canceled)